JAIST Repository
https://dspace.jaist.ac.jp/

| | |
|---|---|
| Title | Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text |
| Author(s) | Pham, Minh Quang Nhat; Nguyen, Minh Le; Shimazu, Akira |
| Citation | Research report (School of Information Science, Japan Advanced Institute of Science and Technology), IS-RR-2013-002: 1-10 |
| Issue Date | 2013-01-18 |
| Type | Technical Report |
| Text version | publisher |
| URL | http://hdl.handle.net/10119/10901 |
| Rights | |
| Description | リサーチレポート（北陸先端科学技術大学院大学情報科学研究科） |

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

# Using Shallow Semantic Parsing and Relation Extraction for Finding Contradiction in Text

Minh Quang Nhat Pham, Minh Le Nguyen and Akira Shimazu
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292, JAPAN
minhpqn,nguyenml,shimazu@jaist.ac.jp

**Abstract**

The problem of text representation is an important issue in textual inference tasks. Given the fact that full predicate-logic analysis is not practical in wide-coverage semantic processing, using shallow semantic representations is an intuitive and straightforward approach. Previous work on finding contradiction in text incorporate information derived from predicate-argument structures as features in supervised machine learning frameworks. In contrast to previous work, we explore the use of shallow semantic representations for contradiction detection in a rule-based framework. We address the low-coverage problem of shallow semantic representations by using a backup module which relies on binary relations extracted from sentences for contradiction detection. Evaluation experiments conducted on standard data sets indicated that using the backup module increases the coverage of contradiction phenomena for the contradiction detection system. Our system achieves better recall and F1 score for contradiction detection than most of baseline methods, and the same recall as a state of the art supervised method for the task.

## 1 Introduction

Finding contradiction in text is a fundamental problem in natural language understanding (De Marneffe et al., 2008). Contradiction detection (CD) is necessary for many potential applications. For instance, contradictions need to be recognized by question answering systems or multi-document summarization systems (Harabagiu et al., 2006). This study addresses the problem of detecting whether the contradiction relationship exists in a pair of a text $T$ and a hypothesis $H$.

To the best of our knowledge, the first systematic investigation of the CD task is the work of Harabagiu et al. (2006), which presented a framework for detecting contradiction phenomena that originate when using (i) negation; (ii) antonymy; or (iii) semantic and pragmatic information. The proposed framework adopted a supervised machine learning approach to recognizing contradiction. De Marneffe et al. (2008) proposed a definition of contradiction for NLP tasks, built corpora and constructed a typology of contradiction classes. They employed supervised machine learning techniques for the task and extracted many contradiction features such as polarity features, number, date and time features.

Supervised machine learning-based frameworks (Harabagiu et al., 2006; De Marneffe et al., 2008) perform well when a training data set which covers many contradiction phenomena is available. However, constructing such a training data set requires much time and human effort because of the complicated nature of contradiction phenomena.

Beyond string-based matching approaches, one can approach to the CD task by applying logical inference techniques. Although the logical inference approach may obtain good precision, it is not widely used for the task due to the fact that full predicate-logic analysis is currently not practical for wide-coverage semantic processing (Burchardt et al., 2009). Given that fact, Burchardt et al. (2009) pointed out that using shallow semantic representations based on predicate-argument structures and frame knowledge is an intuitive and straightforward approach to textual inference tasks.

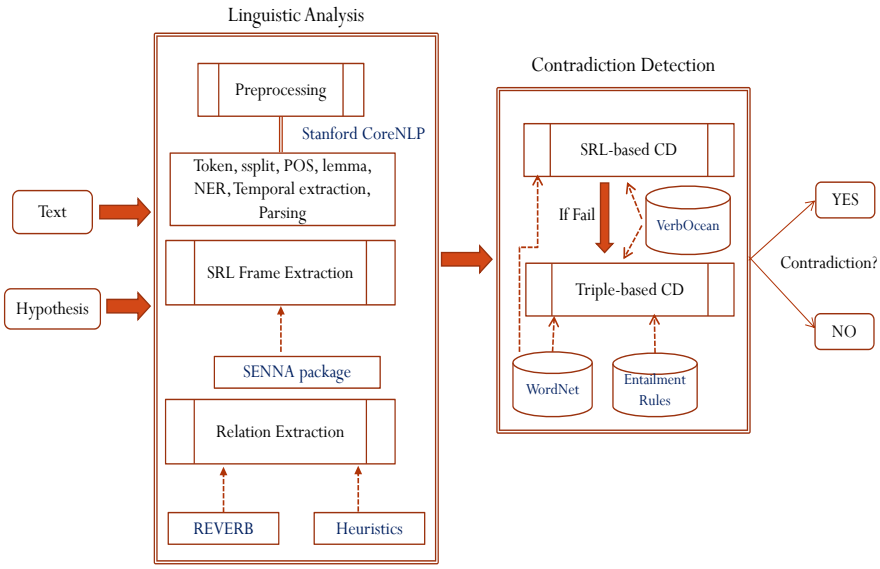


Figure 1: The architecture of the contradiction detection system

Previous work on contradiction detection integrate predicate-argument structures and frame knowledge as features in machine learning-based systems. Harabagiu et al. (2006) partially used features derived from the alignment of predicates and arguments across the text and the hypothesis. De Marneffe et al. (2008) used structural features which capture the role exchange between the subjects in the text and the objects in the hypothesis for aligned verbs.

In contrast to previous work on the CD task, we propose a novel rule-based system for finding contradiction in text. The main component of our system is a contradiction detection module which relies on the alignment of semantic role (SRL) frames extracted from the text and the hypothesis in each pair. We define a contradiction measurement based on that alignment. The main limitation of using semantic role knowledge for the task is the low coverage of semantic role resources and errors propagated from automatic SRL systems. We address those problems by using a backup CD module which performs contradiction detection over binary relations extracted from the text and the hypothesis. If the SRL-based module fails to identify the contradiction relationship in the pair, the second module will be applied. Evaluation experiments on standard data sets obtained from RTE challenges (Giampiccolo et al., 2007, 2008; Bentivogli et al., 2009) show that the proposed system achieves better recall and F1 score for contradiction detection than most of baseline methods, and the same recall as a state of the art supervised method for the task. Furthermore, experimental results also indicate that using the backup module increases the coverage of contradiction phenomena for the system.

## 2 Overview of the Proposed Contradiction Detection System

Figure 1 shows the architecture of the proposed system. The system takes as input a pair *(T,H)*. First, *T* and *H* are input to the Linguistic Analysis module. The Linguistic Analysis module performs text preprocessing, semantic role labeling (SRL), and relation extraction for *T* and *H*. Next, in the contradiction detection component, we combine two CD modules. The first module – SRL-based module checks the contradiction relationship in the pair over verb frames (SRL frames), and the second module – triple-based module utilizes binary relations extracted from *T* and *H* for classification. The CD component is organized in a two-stage scheme. If the SRL-based module fails to check the contradiction relationship, the triple-based module will be used as a backup engine. The two-stage scheme is proposed to address the low-coverage problem of the SRL-based module. Technical details of the two CD modules of the CD component are described in the next following sections.

Table 1: SRL frames extracted from the sentence "Bell, a company which is based in LA, makes and distributes computer products."

| Verb | Element List |
|---|---|
| based | A1: "a company" |
| | R-A1: "which" |
| | AM-LOC: "in LA" |
| makes | A0: "Bell, a company which is based in LA," |
| | A1: "computer products" |
| distributes | A0: "Bell, a company which is based in LA," |
| | A1: "computer products" |

# 3 Linguistic Analysis

## 3.1 Extracting SRL Frames

After performing text preprocessing by using an off-the-shelf NLP software, Stanford CoreNLP [1], we utilize SENNA package[2] (Collobert et al., 2011) for semantic role labeling. SENNA is a robust semantic role labeling system with relatively good accuracy. On CoNLL 2005 data, it achieves 75.49% of F1 score. After performing tokenization and sentence segmentation on a text segment by using Stanford CoreNLP, the preprocessed text segment is input to the SENNA system for shallow semantic analysis. Then, from the output of SENNA, we extract a set of SRL frames. Table 1 shows SRL frames extracted from an example sentence. An SRL frame consists of a verb predicate and a list of SRL elements. Types of SRL elements follow notations used in PropBank (Palmer et al., 2005).

## 3.2 Binary Relation Extraction

The triple-based CD module takes as input two sets of relations extracted from *T* and *H*. In our system, we extract binary relations in the form of a triple (*arg1*, *R*, *arg2*), in which *R* represents the relation phrase between two arguments: *arg1* and *arg2*. For instance, the triple ("John", "was born in", "Canada") is extracted from the sentence "John was born in Canada."

### Extracting triples by using REVERB

REVERB is a tool which can automatically identify and extract binary relations from English sentences. The input of REVERB is a POS-tagged and NP-chunked sentence and its output is a set of extraction triples. In order to provide information of how reliable an extraction triple is, REVERB assigns confidence scores for resulting extraction triples by using a logistic regression classifier. The confidence function is trained on manually annotated extraction triples extracted from 1000 sentences from the Web and Wikipedia. In this study, we only use high-score extraction triples.

Although triples extracted by REVERB are useful for the CD task, there are many useful relations that REVERB cannot extract. First, in a triple extracted by REVERB, arguments are nearest noun phrases to the right and the left of the relation phrase, so relations between noun phrases whose distances are long may not be recognized, such as the equivalent relation between two entity mentions in the same co-reference chain. Analyzing contradiction examples in data sets of RTE competitions, we find that "isA" relations which specify the equivalent relation of two objects, are useful relations for the CD task. Second, in some cases, relation phrases of two extraction triples cannot be compared without using inference rules that specify the entailment relationship between two triples. Thus, we use the available corpus of inference rules obtained from (Berant et al., 2011) to transform original extraction triples.

[1] Stanford CoreNLP is available online on: http://nlp.stanford.edu/software/corenlp.shtml

[2] SENNA is available online on: http://ml.nec-labs.com/senna/

**Extracting "isA" relations from co-reference chains**

One key improvement that has been made on data sets of recent RTE challenges is that in each pair, the text *T* is normally a text segment of multiple sentences (Giampiccolo et al., 2008). Thus, co-reference resolution is an useful information source for RTE. Given a co-reference chain $C$ of entity mentions referring to the same entity in the world, we apply the procedure as follows to obtain "isA" relations. First, we extract the set of mentions which are recognized as named entities in the chain $C$. Denote the named entity set as $C_1$ and the set of remaining mentions in the chain $C$ as $C_2$. For each mention $M_1$ in $C_1$ and mention $M_2$ in $C_2$, we generate the "isA" relation $(M_1, isA, M_2)$.

**Extracting "isA" relations from noun phrases**

The second source from which we extract "isA" relations is noun phrases. If the ending part a noun phrase NP is recognized as a named entity, we can extract an "isA" relation from that. For example, the triple ("Peter Lawrence", isA, "her father") is extracted from the noun phrase "Her father Peter Lawrence." In order to avoid incorrect triples to be extracted like the triple ("John and", isA, Mary) from the noun phrase "John and Mary," we only extract "isA" relations that satisfy three following constraints: i) the first argument is an LOCATION, ORGANIZATION, or PERSON entity; ii) the second argument must include at least one noun; and iii) the ending word of the second argument is not a conjunction word such as "and", "or", or "nor".

**Extracting "isA" relations from "abbrev" relations in dependency parses**

In typed-dependency outputs of CoreNLP, an abbreviation modifier of an NP is a parenthesized NP that servers to abbreviate the NP (or to define an abbreviation). For example in a text fragment "Niger Justice Movement (MNJ)", the "abbrev" dependency relation is *abbrev*(Movement, MNJ). From that "abbrev" relation, we can extract the triple (MNJ, isA, Movement). However, the desired extraction triple is (MNJ, isA, "Niger Justice Movement"). In order to obtain such an extraction triple, we propose a heuristic algorithm as follows. Starting from the head node "Movement" in the relation, the algorithm goes back through its previous tokens until "noun compound modifiers" (nn), "adjectival modifiers" (amod), or "determiner modifiers" (det) of the head node cannot be found. In our case, tokens "Niger" and "Justice" are "nn" modifiers of "Movement". After finding the position of the head's last modifier, we obtain the second argument of the desired triple by extracting token sequence from that position to the head node. In the example, we obtain the sequence "Niger Justice Movement." The procedure to identify the first argument is similar.

**Transforming triples using entailment rules**

Entailment rules or inference rules which specify directional entailment relations between two text fragments have been shown to be useful for RTE and question answering (Berant et al., 2011). For instance, the rule "X purchase Y $\rightarrow$ X acquire Y" helps to recognize that the text "Google purchased reMail" answers the question "Which company acquired reMail?." Berant et al. (2011) presented a method for learning typed entailment rules from a large data set. Extracted rules are in the forms "X::predicate$_1$::Y $\rightarrow$ X::predicate$_2$::Y", X::predicate$_1$::Y $\rightarrow$ Y::predicate$_2$::X, or X::predicate::Y $\rightarrow$ Y::predicate::X."

In this study, we use the corpus of $30,000$ entailment rules between typed predicates, which is obtained from (Berant et al., 2011) for transforming triples generated by REVERB into entailed triples. Transformed triples are potentially useful in recognizing contradiction of triple pairs in which other semantic resources like WordNet do not cover the relationship between their predicates. The procedure for transforming a triple by using the entailment rule corpus is as follows. Given a triple $(x, r, y)$, we search for rules in the entailment rule corpus such that predicates of their left-hand-side triples match the relation phrase $r$. Then, we use found rules to transform the triple $(x, r, y)$. Since several rules can be found, multiple entailed triples may be generated.

# 4 Contradiction Detection by Matching Semantic Frames

## 4.1 Notation

An SRL frame is denoted by a tuple $S = \{V, E_1, \ldots, E_k\}$, where $V$ is used to denote the verb predicate; and $E_i$ represents the $i$-th SRL element in the frame. Each SRL element is represented in the form $\{\mathcal{T}, \{N\}\}$ in which $\mathcal{T}$ specifies the type of the SRL element, and $\{N\}$ is the set of underlying tokens of the element. Types of SRL elements follows the annotation guideline in PropBank (Palmer et al., 2005). SRL elements can be arguments or modifiers (adjuncts).

A text segment consists of a set of SRL frames. We denote two sets of SRL frames of *T* and *H* by $T = \{S_i^{(t)}\}_{i=1}^{m}$ and $H = \{S_j^{(h)}\}_{j=1}^{n}$, in which $m$ and $n$ are the number of SRL frames extracted from *T* and *H*, respectively.

## 4.2 Contradiction Detection Model

The contradiction detection model consists of a contradiction function $\mathcal{F}_S(T, H)$ which calculates the contradiction measurement for the pair (*T*, *H*) on their extracted SRL frames. Then, the score computed by $\mathcal{F}_S(T, H)$ is compared with a threshold value $t_1$. If $\mathcal{F}_S(T, H) \geq t_1$, we determine that *T* and *H* are contradictory. If $\mathcal{F}_S(T, H) < t_1$ and $\mathcal{F}_S(T, H) \neq \mathfrak{F}$, we determine that *T* and *H* are not contradictory. Here, we use a special value $\mathfrak{F}$ to indicate that the contradiction relationship in the pair cannot be recognized by only using SRL frames. For instance, the SRL-based module cannot detect the contradiction relationship in a pair if there is no SRL frame extracted from *T* or *H*.

In order to define the contradiction function $\mathcal{F}_S(T, H)$, we utilize the observation that *T* and *H* are contradictory if there exists an event indicated by an SRL frame in *H*, which is incompatible with an event indicated by *T*. Formally, the function $\mathcal{F}_S(T, H)$ is defined as following:

$$\mathcal{F}_S(T, H) = \max_{S_i^{(t)} \in T, S_j^{(h)} \in H} \mathbf{f}(S_i^{(t)}, S_j^{(h)}), \quad (1)$$

where $S_i^{(t)}$ and $S_j^{(h)}$ are two SRL frames in *T* and *H*, respectively; and $\mathbf{f}(S_i^{(t)}, S_j^{(h)})$ is a contradiction function defined on the two SRL frames. In natural language, the contradiction measurement of a pair (*T*, *H*) is defined as the maximum contradiction score over all scores of possible pairs of an SRL frame in *T* and an SRL frame in *H*.

Next, we define the function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ of two SRL frames $S_1^{(t)} \in T$ and $S_2^{(t)} \in H$. For concreteness, we denote $S_1^{(t)} = \{V_1, E_1^{(1)}, \ldots, E_k^{(1)}\}$ and $S_2^{(h)} = \{V_2, E_1^{(2)}, \ldots, E_\ell^{(2)}\}$.

The function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ relies on the alignment of SRL elements across two frames. Since the number of SRL elements in an SRL frame is not very large, we propose a greedy alignment algorithm that considers all possible pairs of an SRL element in $S_1^{(t)}$ and an SRL element in $S_2^{(h)}$.

The alignment process is divided into two steps. In the first step, we construct a bipartite graph that stores similarity scores of all possible pairs of an SRL element in $S_1^{(t)}$ and an SRL element in $S_2^{(h)}$. The semantic similarity of two SRL elements is computed by applying the local lexical level matching method (Dagan et al., 2007). We utilize co-reference resolution information in computing element similarity by substituting mentions found in an SRL element with their equivalent mentions in the corresponding co-reference chain. In the second step, from the similarity graph, we construct an alignment graph for SRL elements across the two SRL frames. The alignment process is done by a greedy algorithm. For each element $E_i^{(1)} \in S_1^{(t)}$, we search for the aligned element $E_j^{(2)} \in S_2^{(h)}$ having the maximum similarity score with $E_i^{(1)}$, and also satisfying $Sim(E_i^{(1)}, E_j^{(2)}) > minValue$. We use a threshold $minValue$ to avoid element pairs that have too low similarity to be aligned. In practice, we choose $minValue = 0.2$.

If the corresponding aligned element of an SRL element $E_i^{(1)} \in S_1^{(t)}$ cannot be found by using the greedy algorithm, the element $E_j^{(2)}$ with the same type as the type of $E_i^{(1)}$ will be chosen. Since in our system we utilize the mismatch of subjects, objects, and modifiers, we separate two kinds of elements

in the alignment process: i) elements whose types are of A0 (subject), A1 (agent, direct object), or A2 (indirect object); and ii) other argument types and modifiers. For example, we do not want an A0-typed element to be aligned with an AM-TMP element (temporal modifier). Besides that, in alignment, we restrict that an SRL element can only be aligned with at most one SRL element. Once the alignment of SRL elements across two frames is generated, the mismatches of aligned elements will be taken into account as cues for contradiction detection.

The contradiction function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is defined in three cases as follows.

**Case 1: Two SRL frames are not related**

In this case, we assign $\mathbf{f}(S_1^{(t)}, S_2^{(h)}) = 0$. The rationale is that two events are not contradictory if they are not related. In order to determine whether two SRL frames are related, we take into account the relatedness of their verb predicates and SRL elements. Formally, the relatedness of two SRL frames is computed by:

$$Relatedness(S_1^{(t)}, S_2^{(h)}) = Relatedness(V_1, V_2) \times max_{i,j} Relatedness(E_i^{(1)}, E_j^{(2)}), \quad (2)$$

where $E_i^{(1)} \in S_1^{(t)}$ and $E_j^{(2)} \in S_2^{(h)}$ are SRL elements; $V_1$ and $V_2$ are verbs of $S_1^{(t)}$ and $S_2^{(h)}$, respectively.

The relatedness of two verbs is assigned to $1.0$ if their relation is found in WordNet (Fellbaum, 1998). In this study, we utilized synonym, hypernym, hyponym, and antonym relations in WordNet. If the relation of two verbs is not found in WordNet, we use VerbOcean database (Chklovski and Pantel, 2004) to obtain their relatedness. In other cases, we employ WordNet::Similarity package (Pedersen et al., 2004) to compute the similarity of two verbs. The relatedness of two SRL elements $E_i^{(1)}$ and $E_j^{(2)}$ is defined as the local lexical level matching score.

The relatedness of two SRL frames is compared with a threshold. If it is below the threshold, then $S_1^{(t)}$ and $S_2^{(h)}$ are not related. In practice, we choose $0.2$ as the relatedness threshold.

**Case 2: Two verb predicates are matching**

Two verbs are matching if they satisfy one of the following criteria: i) they have the same surface or base form; ii) they are synonyms in WordNet; or iii) their WordNet-based semantic similarity is not less than a predefined threshold. In practice, we choose $0.85$ as the threshold.

The function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is defined based on the alignment generated in the alignment process. We consider three possible cases as follows.

(a) *All arguments with types A0, A1, or A2 in $S_2^{(h)}$ are aligned and matched with same-type arguments in $S_1^{(t)}$*. Here, two SRL elements are matching if their similarity score is not less than a constant threshold. In practice, we use $0.7$ as the threshold.

In this case, if the contradiction relation exists in the pair, it potentially is triggered by the incompatibility of other arguments and modifiers such as temporal, location, or negation modifiers. Thus, the function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is assigned to $1.0$ if there is any mismatch in temporal, location, or negation modifiers. In other cases, $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is assigned to the maximum contradiction score over all aligned element pairs. The contradiction score of two modifier-typed SRL elements, $E^{(1)}$ and $E^{(2)}$ is defined as $0.5 \times (1 - Sim(E^{(1)}, E^{(2)}))$ where $Sim(E^{(1)}, E^{(2)})$ is the similarity of two elements. The coefficient $0.5$ is used to reduce false-positive predictions, because in our observation, the difference of other modifiers is a less concrete contradiction evidence than that of temporal or location modifiers.

(b) *All arguments with types A0, A1, or A2 in $S_2^{(h)}$ are aligned with same-type arguments in $S_1^{(t)}$; and subjects (A0) are matching (if any), but there exists some mismatched aligned arguments*.

We determine mismatched aligned arguments in the order A1, A2. If any mismatched aligned pair is found, $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is assigned to the contradiction score of that argument pair. The contradiction score of two arguments $E^{(1)}$ and $E^{(2)}$ is defined as $1 - Sim(E^{(1)}, E^{(2)})$.

(c) *Some arguments with types A0, A1, or A2 in $S_2^{(h)}$ are aligned to type-different arguments in $S_1^{(t)}$*.

This case captures the intuition that the contradiction relationship can be realized in the form of the exchange between subjects and objects in two SRL frames. The function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ will be assigned to $1.0$ if that phenomenon is recognized.

**Case 3: Two verb predicates are opposite**

Two verbs are opposite if they are found as antonym verbs in WordNet or opposite verbs in VerbOcean. In this case, the contradiction function $\mathbf{f}(S_1^{(t)}, S_2^{(h)})$ is defined as the similarity of their SRL elements. We define the element-based similarity of two frames as the product of similarity scores of the aligned elements having the same type. The similarity score between two SRL elements is calculated in the alignment step.

# 5 Contradiction Detection by Relation Matching

The main idea of this module is as follows. In the first step, we extract triples from *T* and *H* by using REVERB tool and our heuristics. Next, we compare each triple in *H* with every triple in *T*, and determine whether the contradiction relationship exists in some pairs of triples. In the module, two kinds of contradiction measurements are calculated: one is based on triples extracted by REVERB tool and the other one is based on triples extracted by our heuristics. For simplicity, in description of the method, we use the same notations for both kinds of contradiction measurements.

Formally, we denote a extraction triple by $(x, r, y)$ where $x$ and $y$ respectively represent the first and second argument, and $r$ represents the relation phrase of the triple. Both arguments $x$ and $y$ have underlying words in the text segment from which they are extracted. If the triple is extracted by REVERB tool, $r$ has underlying words in the text segment. If the triple is extracted by our heuristic methods, $r$ does not have underlying words in the text segment.

The text and the hypothesis consist of sets of triples. We denote $T = \{(x_i^{(t)}, r_i^{(t)}, y_i^{(t)})\}_{i=1}^{m}$ and $H = \{(x_j^{(h)}, r_j^{(h)}, y_j^{(h)})\}_{j=1}^{n}$. Here, $m$ and $n$ are respectively the numbers of triples in *T* and *H*. The contradiction detection task is reduced to searching for incompatible triple pairs across *T* and *H*. We define the contradiction function on triples of *T* and *H* as follows.

$$\mathcal{F}_T(T, H) = \max_{T_i \in T; H_j \in H} \mathbf{g}(T_i, H_j), \tag{3}$$

where $T_i$ is the *i*-th triple of *T*; $H_j$ is the *j*-th triple of *H*; and $\mathbf{g}(T_i, H_j)$ is the contradiction function of the two triples $T_i$ and $H_j$.

Due to the limitation of space, we omit the technical details of the procedure for calculating the function $\mathbf{g}(T_i, H_j)$, and only present the main points of the procedure. The function is based on the mismatch of two triples $T_i$ and $H_j$. We consider three cases as follows. If their relation phrases and first arguments are matching, the mismatch of second arguments will be calculated. If two relation phrases are matching and roles of arguments in the two triples are exchanged, $\mathbf{g}(T_i, H_j)$ is assigned to $1.0$. However, this rule is not applied for "isA" (equivalent) relations. In contrast, if two relation phrases are opposite, the similarity measures of first arguments and second arguments are taken into account.

In the procedure for calculating $\mathbf{g}(T_i, H_j)$, we need to determine whether two relation phrases $r_i^{(t)}$ and $r_j^{(h)}$ are matching or not. If the surface and base forms of two relation phrases are different, we use WordNet to detect whether main verbs of $r_i^{(t)}$ and $r_j^{(h)}$ are synonyms. In order to check if two relation phrases $r_i^{(t)}$ and $r_j^{(h)}$ are opposite or not, we utilize antonym relations in WordNet and opposite relations in VerbOcean.

In the module, that two arguments are matching is checked by using their similarity. The similarity score of two arguments is computed by the same method as that for computing the similarity of two SRL elements. When we detect the contradiction of two arguments, we use the contradiction rule as follows. Two arguments are contradictory if they include two entities having the same type but different values.

Table 2: Label distribution in three test sets

| Data Set | Contradiction | Entailment | Unknown | Total |
|---|---|---|---|---|
| RTE-3 Test | 72 | 410 | 318 | 800 |
| RTE-4 Test | 150 | 500 | 350 | 1000 |
| RTE-5 Test | 90 | 300 | 210 | 600 |

Especially, we take into account four categories: NUMBER, DATE, TIME, and LOCATION. In other cases, we use the difference of two arguments as the evidence for contradiction detection.

# 6 Evaluation Experiments

## 6.1 Data Sets

In experiments, we evaluate the proposed method on the test sets of the three-way subtask at RTE-3, RTE-4, and RTE-5 competitions (Giampiccolo et al., 2007, 2008; Bentivogli et al., 2009). The development sets provided at each competition are used to tuned threshold values in two CD modules of the system.

The first two editions of RTE challenges focused on the binary classification setting of RTE, i.e. the task is to classify whether a pair of two text portions is entailment or non-entailment. From RTE-3 challenge, the three-way subtask was proposed. The three-way subtask requires participant systems to decide whether the entailment, contradiction, or independent (unknown) relationship exists in a pair. Since in this study, we focus on contradiction relationship in a text pair, entailment and unknown labels in data sets are converted into non-contradiction labels. The contradiction is rare in data sets of RTE challenges. Table 2 provides statistics on the test sets of three-way subtask in RTE-3, RTE-4, and RTE-5.

The data sets used in experiments are unbalanced, so the average accuracy over all labels is not an appropriate evaluation measures. Therefore, we use Precision, Recall, and F1 score of the contradiction label as evaluation measures.

## 6.2 Baseline Methods

The first baseline method is the method presented in (De Marneffe et al., 2008), which employed supervised machine learning techniques for the CD task. To the best of our knowledge, (De Marneffe et al., 2008) is the only contradiction detection-focused work that evaluates on data sets of RTE challenges.

The second baseline is the BLUE system of Boeing's team (Clark and Harrison, 2009) at RTE-4 and RTE-5 competitions. The BLUE system adopted the logical inference approach to RTE, which performs inference on logic-based representations of the text and the hypothesis in a pair. The reason why we use this baseline is that both our system and the BLUE system use semantic representations of sentences for reasoning. We use best scores among submitted runs of the BLUE system at each competition.

In experiments, we also compare the results achieved by our system with average results of submitted systems for three-way subtask at RTE-3, RTE-4 and RTE-5 challenges. The numbers submitted systems in RTE-3, RTE-4 and RTE-5 for the three-way subtask are 12, 34, and 24 submissions, respectively.

In order to assess the effectiveness of the two-stage system scheme, we separately run each CD module on the three data sets and compare the results with those of the combined system. The first module will compare the SRL-based contradiction score of each pair with a threshold. If the score is greater than or equal to the threshold, it determines that the contradiction relation exists in the pair. Similarly, the second module recognizes the contradiction relationship by using triple-based contradiction scores which are calculated on the pair.

## 6.3 Experimental Results

Table 3 provides experimental results achieved on test sets of RTE-3, RTE-4, and RTE-5 challenges by our system and baseline methods. As shown in results, the proposed system consistently obtained better

Table 3: Experimental results on three data sets

| Method | RTE-3 Pilot | | | RTE-4 Test | | | RTE-5 Test | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| De Marneffe (2008) | 22.95 | 19.44 | 21.04 | – | – | – | – | – | – |
| BLUE system | – | – | – | 41.67 | 10.0 | 16.13 | 42.86 | 6.67 | 11.54 |
| Average result | 10.72 | 11.69 | 11.18 | 25.26 | 13.47 | 13.63 | 26.40 | 13.70 | 14.79 |
| SRL-based | 13.41 | 15.27 | 14.28 | 22.41 | 17.33 | 19.55 | 22.72 | 16.67 | 19.23 |
| Triple-based | 22.58 | 9.72 | 13.59 | 26.3 | 10.0 | 14.49 | 19.48 | 16.67 | 17.96 |
| **Two-stage (our system)** | **14.0** | **19.44** | **16.27** | **23.0** | **22.67** | **22.82** | **21.14** | **28.89** | **24.4** |

recall values and F1 scores than those of baseline methods except the supervised machine learning-based method in (De Marneffe et al., 2008). The BLUE system obtained good prevision but much lower recall values than those achieved by our system. It indicated that our system can recognize more contradiction phenomena than most of baseline methods. Compare with the method presented (De Marneffe et al., 2008), our system achieves the same recall but lower precision. However, the method in (De Marneffe et al., 2008) requires manually annotated training data of contradiction examples.

The results shown in Table 3 indicated that the SRL-based module consistently achieved better recall and F1 score than those of the triple-based module. A possible explanation is that the information contained in shallow semantic representations is richer than that of extraction triples, so the SRL-based module covers more contradiction phenomena than the triple-based module. As expected, the combined system consistently obtained better recall and F1 score than each separate module. Experimental results confirmed our observation that the second backup module increases the coverage of contradiction phenomena for our system.

## 6.4 Error Analysis

In order to better understand the limitations of the proposed system, we analyse some typical incorrect predictions made by our system. We find that many unsuccessful cases are due to that our system does not take into account contradiction phenomena triggered by unary relations. Let us consider the contradiction pair 28 in RTE-4 test set as follows.

**Text:** Lower food prices pushed the UK's inflation rate down to 1.1% in August, the lowest level since 1963. The headline rate of inflation fell to 1.1% in August, pushed down by falling food prices.

**Hypothesis:** Food prices are on the increase.

In the pair 28, contradiction relation triggered by the incompatibility of unary relations ("Food prices", "lower") and ("Food prices", "falling") from the text; and ("Food prices", "on the increase") from the hypothesis. However, our current system does not exploit unary relations like that. We plan to address that issue in the future work.

The second limitation of the proposed method is the lack of common sense knowledge. For instance, consider the pair 909 extracted from RTE-4 test set as follows.

**Text:** Morales' left of center policies, especially his support for the coca industry will likely not make him a popular Latin American figure for the Bush administration in the United States, who might be afraid of a closer alliance between Morales, Hugo Chvez in Venezuela, Fidel Castro of Cuba, and even the more moderate but still left-of-center Luiz Lula da Silva of Brazil.

**Hypothesis:** The Bush administration supports Morales.

In the pair 909, common sense knowledge is needed to know that the text implies that "the Bush administration does not support Morales" because the Bush administration "might be afraid of a closer alliance between Morales, Hugo Chvez in Venezuela, Fidel Castro of Cuba". Our current contradiction detection system does not incorporate such common sense knowledge, so it cannot correctly recognize the contradiction relationship of the pair.

## 7 Conclusion

In this paper, we have presented a new rule-based method for finding contradiction in text. We define contradiction measurements on the predicate-argument structures and binary relations extracted from the text and the hypothesis in a pair. We deal with the low-coverage problem of semantic role resources by using a backup module which exploits extraction triples. Current off-the-shelf relation extraction systems miss many useful relations for contradiction detection. Thus, we have proposed several heuristics for extracting additional relations from different text representations such as noun phrases or typed-dependency trees. Experimental results achieved on standard data sets showed that our proposed system obtained better recall and F1 score for contradiction detection than most of baseline methods.

## References


Bentivogli, L., I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini (2009). The fifth pascal recognizing textual entailment challenge. In *In Proceedings of TAC Workshop*.

Berant, J., I. Dagan, and J. Goldberger (2011, June). Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, pp. 610–619. Association for Computational Linguistics.

Burchardt, A., M. Pennacchiotti, S. Thater, and M. Pinkal (2009). Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering 15*(Special Issue 04), 527–550.

Chklovski, T. and P. Pantel (2004, July). Verbocean: Mining the web for fine-grained semantic verb relations. In D. Lin and D. Wu (Eds.), *Proceedings of EMNLP 2004*, Barcelona, Spain, pp. 33–40. Association for Computational Linguistics.

Clark, P. and P. Harrison (2009). Recognizing textual entailment with logical inference. In *In Proceedings of the First Text Analysis Conference (TAC 2008)*.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011, November). Natural language processing (almost) from scratch. *J. Mach. Learn. Res. 999888*, 2493–2537.

Dagan, I., D. Roth, and F. Massimo (2007). A tutorial on textual entailment.

De Marneffe, M.-c., A. N. Rafferty, and C. D. Manning (2008). Finding contradictions in text. In *In Proceedings of ACL 2008*.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Giampiccolo, D., H. T. Dang, B. Magnini, I. Dagan, E. Cabrio, and B. Dolan (2008). The fourth pascal recognizing textual entailment challenge. In *In Proceedings of TAC 2008 Workshop*.

Giampiccolo, D., B. Magnini, I. Dagan, and B. Dolan (2007). The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 1–9.

Harabagiu, S., A. Hickl, and F. Lacatusu (2006). Negation, contrast, and contradiction in text processing. In *Proceedings of the Twenty-First Na-tional Conference on Artificial Intelligence (AAAI-06)*.

Palmer, M., D. Gildea, and P. Kingsbury (2005, March). The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist. 31*(1), 71–106.

Pedersen, T., S. Patwardhan, and J. Michelizzi (2004). Wordnet::similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, Stroudsburg, PA, USA, pp. 38–41. Association for Computational Linguistics.